REMARKS/ARGUMENTS

Claims 1, 11 and 20 are pending in the present application and were amended. No claims were added or canceled. Support for the amendments can be found, for example, on page 34 of the specification. Reconsideration of the rejection is respectfully requested in view of the above amendments and the following comments.

I. 35 U.S.C. § 103, Obviousness: Claims 1, 11, and 20

The Examiner has rejected claims 1, 11, and 20 under 35 U.S.C. 103(a) as being unpatentable over "IBM Hardware Performance Monitor (hpm)", August 2002 (art made of record, hereinafter referred to as "IBM-HPM") in view of US Patent Publication No. 2003/0005422 A1 to Kosche et al. (art made of record, hereinafter referred to as "Kosche") and further in view of US Patent No. 6,681,388 to Sato et al. (art made of record, hereinafter referred to as "Sato"). This rejection is respectfully traversed.

In rejecting the claims, the Examiner states:

> **Claim 1:**
>
> IBM-HPM discloses a computer program product, a system, and *a method, in a data processing system, for optimizing runtime execution of a computer program* (e.g., pages .3, 8), *comprising*:
>
> *obtaining performance profile data accumulated during a trace of a computer program execution to include annotations based on the occurrence of one or more events during execution of the computer program* (e.g., pages 3, 6), *wherein the one or more events occur based on hardware counter values and performance indicators associated with one or more portions of the computer program* (e.g., pages 4-5);
>
> *obtaining code for the computer program* (e.g., pages 27-28); *determining a manner for compiling the code to provide one or more optimizations to the runtime execution of the computer program based on the performance profile data and Me annotations* (e.g., pages 27-29),
>
> *wherein the one or more optimizations include at least one of an optimization to branch prediction* (e.g., pages 10, 12), and
>
> *an optimization to cache misses and cache hit rate* (e.g., pages 6, 10, 12);
>
> *presenting the one or more optimizations to a programmer using one or more graphical user interfaces* (e.g., page 27-28);
>
> *receiving one or more selected optimizations of the one or more optimizations selected by the programmer* (e.g., pages 27-28); and
>
> *compiling the code using the one or more .selected optimizations to generate an optimized computer program* (e.g., page 27, 16, 21,26, and 32).
>
> IBM-HPM discloses profiling and potimizing branch prediction and branch misprediction rate (pages 10, 12), but does not explicitly discloses *one or more optimizations include at least one of an optimization to instruction paths of the computer program at branch points such that a more contiguous execution of instructions within the computer program is achieved.*

However, in an analogous art, Kosche further discloses *one or more optimizations include at least one of an optimization to instruction paths of the computer program at branch points such that a more contiguous execution of instructions within the computer program is achieved* (e.g., [0029]; [0034-0036]; FIG. 4-6 and related text).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Kosche's teaching into IBM-HPM's teaching. One would have been motivated to do so to make operations faster, executing more operations in parallel, increase performance, and avoid pipeline stalls as suggested by Kosche (e.g., [0007-0008] and [0011-0020]).

IBM-HPM discloses profiling and optimizing cache misses and cache hit rate (pages 6, 10, 12) but neither IBM-HPM nor Kosche explicitly discloses one or more optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines.

However, in an analogous art, Sato further discloses one or more optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines (e.g., col.29: 1-29; col.1: 29-46; FIG. 27-29 and related text).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sato's teaching into IBM-HPM and Kosche's teaching. One would have been motivated to do so to reduce inter-cache conflict and shorten the execution time of the program as suggested by Sato (e.g., col.2: 18-22).

Claim 11:

Claim 11 is a computer program product version, which recites the same limitations as those of claim 1, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claim, they also teach all of the limitations of claim 11.

Claim 20:

Claim 20 is a system version, which recites the same limitations as those of claims 1 and 11, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the references teach all of the limitations of the above claims, they also teach all of the limitations of claim 20.

Office Action dated August 17, 2007, pages 3-5.

Claim 1 as amended herein is as follows:

1. A method, in a data processing system, for optimizing runtime execution of a computer program, comprising:

modifying performance profile data accumulated during a trace of a computer program execution to form annotated performance profile data, wherein the annotated performance profile data includes annotations based on the occurrence of one or more events during execution of the computer program, wherein the one or more events occur based on hardware counter values and on performance indicators being associated with one or more portions of the computer program;

obtaining code for the computer program;

determining a manner for compiling the code to provide one or more optimizations to the runtime execution of the computer program based on the performance profile data and the annotations of the annotated performance profile data, wherein the one or more optimizations include at least one of an optimization to instruction paths of the computer program at branch

points such that a contiguous execution of instructions within the computer program is achieved, and an optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different cache lines;

presenting the one or more optimizations to a programmer for selection using one or more graphical user interfaces;

receiving one or more selected optimizations of the one or more optimizations selected by the programmer; and

compiling the code using the one or more selected optimizations to generate an optimized computer program.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. §103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Additionally, all limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Therefore, no *prima facie* obviousness rejection can be established if the proposed combination does not teach all of the features of the claimed invention.

In the present case, the Examiner has failed to establish a *prima facie* case of obviousness because the references, considered alone or in combination, fail to teach all of the features of claim 1 as amended herein. For example, neither IBM-HPM, Kosche, or Sato or their combination discloses or suggests "modifying performance profile data accumulated during a trace of a computer program execution to form annotated performance profile data, wherein the annotated performance profile data includes annotations based on the occurrence of one or more events during execution of the computer program, wherein the one or more events occur based on hardware counter values and on performance indicators being associated with one or more portions of the computer program", or "determining a manner for compiling the code [that is obtained for the computer program] to provide one or more optimizations to the runtime execution of the computer program based on the performance profile data and the annotations of the annotated performance profile data" as recited in claim 1.

The Examiner refers to pages 3 and 6 of IBM-HPM as disclosing obtaining performance profile data accumulated during a trace of a computer program execution to include annotations based on the occurrence of one or more events during execution of the computer program. Applicants respectfully disagree.

IBM-HPM is a description of a hardware performance monitor that can be used to monitor processor level or data processing system level activities. Page 3 lists examples of activities that can be monitored or counted including, for example, instructions, cache misses and memory traffic. Page 6 of IBM-HPM describes that only activities that are needed should be counted, and provides examples such as cycles. loads, stores, cache misses, branches, branch misses, and the like.

Nowhere on page 3 or 6 or elsewhere, however, does IBM-HPM disclose a method for optimizing runtime execution of a computer program that includes "modifying performance profile data accumulated during a trace of a computer program execution to form annotated performance profile data, wherein the annotated performance profile data includes annotations based on the occurrence of one or more events during execution of the computer program". The reference does not discuss modifying performance profile data to provide annotated performance profile data, and does not disclose annotated performance profile data that includes annotations based on the occurrence of one or more events during execution of the computer program as recited in claim 1.

IBM-HPM also does not disclose or suggest "wherein the one or more events occur based on hardware counter values and on performance indicators being associated with one or more portions of the computer program" as additionally recited in claim 1. The Examiner refers to pages 4 and 5 of IBM-HPM as disclosing "wherein the one or more events occur based on hardware counter values and performance indicators associated with one or more portions of the computer program." Applicants respectfully disagree. Page 4 describes when hardware counters may be used (one example is when simulators are not available or are not enough), and page 5 describes advantages of performance tools. Again, there is no disclosure of annotated performance profile data, wherein the annotations are based on the occurrence of one or more events during execution of the computer program, nor is there any disclosure of such events occurring "based on hardware counter values and on performance indicators being associated with one or more portions of the computer program" as now clearly recited in claim 1.

Thus, IBM-HPM does not disclose or suggest "modifying performance profile data accumulated during a trace of a computer program execution to form annotated performance profile data, wherein the annotated performance profile data includes annotations based on the occurrence of one or more events during execution of the computer program, wherein the one or more events occur based on hardware counter values and on performance indicators being associated with one or more portions of the computer program." Furthermore, because IBM-HPM does not disclose forming annotated performance profile data, the reference also does not disclose or suggest "determining a manner for compiling the code to provide one or more optimizations to the runtime execution of the computer program based on the performance profile data and the annotations of the annotated performance profile data" as additionally recited in claim 1 (emphasis added).

Neither Kosche nor Sato supplies the deficiencies in IBM-HPM. Kosche is cited as disclosing "one or more optimizations include at least one of an optimization to instruction paths of the computer program at branch points such that a more contiguous execution of instructions within the computer program is achieved", and Sato is cited as disclosing "one or more optimization to storage of instructions or data in a cache so that portions of a cache line that is falsely shared are stored in the cache on different

cache lines", which the Examiner acknowledges is not disclosed in IBM-HPM. Assuming *arguendo* that Kosche and Sato describe the optimizations as proposed by the Examiner, IBM-HPM in view of Kosche and Sato still fails to disclose or teach "modifying performance profile data accumulated during a trace of a computer program execution to form annotated performance profile data, wherein the annotated performance profile data includes annotations based on the occurrence of one or more events during execution of the computer program, wherein the one or more events occur based on hardware counter values and on performance indicators being associated with one or more portions of the computer program", or "determining a manner for compiling the code [that is obtained for the computer program] to provide one or more optimizations to the runtime execution of the computer program based on the performance profile data and the annotations of the annotated performance profile data" as recited in claim 1.

Yet further, although Kosche may describe improving the prediction rate of unpredictable branches, it does not specifically disclose or suggest "one or more optimizations include at least one of an optimization to instruction paths of the computer program at branch points such that a contiguous execution of instructions within the computer program is achieved" as recited in claim 1.

For at least all the above reasons, neither IBM-HPM, Kosche, or Sato or their combination teaches all of the features of claim 1, and the Examiner has failed to establish a *prima facie* case of obviousness in rejecting claim 1. Claim 1, accordingly, is not obvious over IBM-HPM, Kosche and Sato and patentably distinguishes over the references in its present form.

Claims 11 and 20 have been amended in a similar manner as claim 1 and also patentably distinguish over the references in their present form.

Therefore, the rejection of claims 1, 11, and 20 under 35 U.S.C. 103(a) has been overcome.

II. Conclusion

For at least all the above reasons, this application is believed to be in condition for allowance, and it is respectfully requested that the Examiner so find and issue a Notice of Allowance in due course.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: November 16, 2007

Respectfully submitted,

/Gerald H. Glanzman/
Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants